基于改进贝叶斯优化算法的 CNN 超参数优化方法 *

邓　帅[a, b]

(北京工业大学 a. 北京未来网络科技高精尖创新中心; b. 北京市物联网软件与系统工程技术研究中心, 北京 100124)

摘　要：CNN 框架中，如何对其模型的超参数进行自动化获取一直是一个重要问题。提出一种基于改进的贝叶斯优化算法的 CNN 超参数优化方法。该方法使用改进的汤普森采样方法作为采集函数，利用改进的马尔可夫链蒙特卡洛算法加速训练高斯代理模型。该方法可以在超参数空间不同的 CNN 框架下进行超参数优化。利用 CIFAR-10、MRBI 和 SVHN 测试集对算法进行性能测试，实验结果表明，改进后的 CNN 超参数优化算法比同类超参数优化算法具有更好的性能。

关键词：贝叶斯优化；卷积神经网络；高斯过程；超参数优化

中图分类号：TP183　　doi: 10.3969/j.issn.1001-3695.2018.01.0021

Hyper-parameter optimization of CNN based on improved Bayesian optimization algorithm

Deng Shuai[a, b]

(*a. Beijing Advanced Innovation Center for Future Internet Technology*, *b. Beijing Engineering Research Center for IoT Software & Systems*, *Beijing University of Technology*, *Beijing* 100124, *China*)

Abstract: In the framework of convolutional neural network (CNN) , how to obtain the hyper-parameters of its model automatically is an important and pressing research topic. In this paper, we proposed a hyper-parameter optimization method of CNN based on improved Bayesian optimization algorithm. This method uses the improved Thompson sampling method as the acquisition function. The improved Markov Chain Monte Carlo algorithm is used to accelerate the Gaussian surrogate model. The proposed method can be used to optimize hyper-parameters in frameworks of CNN with different hyper-parameter space. The performance of the algorithm was tested by using these testing sets: CIFAR-10, MRBI and SVHN. The experimental results show that the improved hyper-parameter optimization algorithm of CNN has better performance than the other algorithms.

Key words: Bayesian optimization; convolutional neural networks; Gaussian process; hyper-parameter optimization

0　引言

由于任何一个 CNN 模型都无法对所有数据集进行最佳泛化，因此在将 CNN 应用于新数据集之前，必须先选择一组适当的超参数。CNN 的超参数包括层数，每层隐藏单元的数量，层的激活函数，层的内核大小，网络内这些层的配置等。为新数据集选择新的模型可能是一个耗时且繁琐的任务。被调整的超参数的数量以及每个新的超参数集合的评估时间使得它们在 CNN 模型中的优化尤其困难。超参数对不同 CNN 架构的影响的研究已经显示出复杂的关系，其中在简单 CNN 网络中提供巨大性能改进的超参数在更复杂的体系结构中并不具有相同的效果[1]。这些研究还表明，一个数据集上的结果可能不会转移到具有不同图像属性、先验概率分布、类别数量或训练示例数量的另一个数据集。由于没有明确的公式来选择一组正确的超参数，所以它们的选择通常取决于先前的经验和实验性错误的组合。由于 CNN 算法计算成本高，在传统的平台上可能需要数天时间来进行训练，所以重复的反复实验既是低效的，也不是彻底的。

针对这种未知黑盒函数的优化，贝叶斯优化[2]提供了一个有效的方法，并且已经被证明在许多具有挑战性的优化基准函数上优于其他的最先进的全局优化算法。对于连续函数，通常假定未知函数是从高斯过程（GP）采样的，并且在观察时保持该函数的后验分布。为了选择下一个实验的超参数，可以优化当前最好的结果或者高斯过程置信区间（UCB）的期望增量（EI）[3,4]。Bergstra 等人[5]提出的 TPE（Tree-structured Parzen Estimator Approach）算法和 Snoek 等人[6]提出的 GP EI MCMC 算法已经证明 EI 和 UCB 在许多黑盒函数的全局最优化的函数评估中是有效的。然而 CNN 超参数优化具有与其他黑盒优化问题相区别的特征。首先每一次对 CNN 超参数的评估可能需要一个可变的时间量，不清楚 EI 和 UCB 是否适用于 CNN 的超参数优

收稿日期：2018-01-10；**修回日期：**2018-03-06　　**基金项目：**北京市自然科学基金资助项目（4122010）；国家自然科学基金资助项目（60773186）

作者简介：邓帅（1993-），男，河北黄骅人，硕士研究生，主要研究方向为物联网、深度学习（353007141@qq.com）.

化的函数评估；并且不确定这些优化算法得到的超参数对于具有不同数据模式或结构的域的 CNN 框架是否最优。

本文基于 CNN 框架和已有贝叶斯优化算法，提出一种改进的贝叶斯优化算法对 CNN 超参数进行优化。该方法利用高斯过程对超参数进行建模，利用改进的马尔可夫链蒙特卡洛（improved Markov Chain Monte Carlo, IMCMC）算法加速训练高斯过程模型的训练，即计算高斯代理模型的超参数（长度尺度和协方差振幅等）。并使用改进的汤普森采样（improved thompson sampling,ITS）方法作为采集函数获得下一个采样点并计算 loss 值，然后合并到历史观测集中。这个过程迭代直到获得一组性能良好的超参数。该方法可以在超参数空间不同的 CNN 框架下进行超参数优化。利用 CIFAR-10、MRBI 和 SVHN 测试集对算法进行性能测试，说明了本文提出的方法获得的超参数比 TPE 和 GP EI MCMC 等同类的优化算法得到的超参数性能更好，更稳定。

1　贝叶斯优化算法介绍

对于超参数的优化，可以将这种优化看做是反映泛化性能的未知黑盒函数的优化，并调用针对这些问题开发的算法。这些优化问题与作为训练过程一部分经常遇到低层次目标是不同的：这里函数评估（求值目标函数调用一次）代价很大，因为它们涉及到主要的机器学习算法的完成。在这种函数求值代价大的情况下，希望花费计算时间来更好的选择在哪里寻找最佳参数。在贝叶斯优化中，感兴趣的是在一些有界集合 Λ 上找到函数 $f(\lambda)$ 的最小值，本文将把它作为 $\mathbb{R}$ 的一个子集。使得贝叶斯优化不同于其他程序的是它为 $f(\lambda)$ 构造一个概率模型，然后利用这个模型来决定Λ在哪里去评估函数，同时整合不确定性。基本的思路是使用 $f(\lambda)$ 以前评估中可用的所有信息来学习目标函数的形态，而不是简单地依靠局部梯度和 Hessian 近似。这可以实现通过较少的评估就可以找到复杂非凸函数的最小值，代价是执行更多的计算以确定下一个采样点。因此分为了两个步骤：a）选择一个先验函数来表达关于被优化函数的假设，本文使用的高斯过程，因为其具有灵活易处理的特性；b）选择一个采集函数，用来从后验模型构造一个效用函数，来确定下一个采样点。

将要优化的 CNN 的超参数看做是多维空间的点。超参数的贝叶斯优化通过在超参数 $\lambda\in\Lambda$ 的空间中对损失函数 $f(\lambda)$ 进行一个高斯先验建模来执行。通过验证集 X_V 可以观察到这种损失函数的一些噪声，$f(\lambda)$ 表示为

$$L(m_\lambda, X_V)=\frac{1}{|X_V|}\sum_{(x_i,y_i)\in X_V} l(m_\lambda(x_i), y_i) \tag{1}$$

$$f(\lambda)=L(m_\lambda, X_V)+\varepsilon \tag{2}$$

其中：m_λ 是通过在给定训练数据集 X_T 上运行具有超参数 λ 的卷积神经网络 A 而获得的模型参数，$l(\cdot,y)$ 是目标损失函数。真实的 $f(\lambda)$ 是未知的，只能通过在验证数据集上计算观测噪声来量化。

2　改进的贝叶斯优化算法

2.1　高斯过程

对于建模损失函数，高斯过程(GP)一直被认为是一种方便且强大的模型优化算法。在这里将采用 $f:\Lambda\to\mathbb{R}$ 的形式。GP 由下列性质定义：任意有限的 N 个点 $\{\lambda_n\in\Lambda\}_{n=1}^N$ 在 $\mathbb{R}^N$ 上引起多变量高斯分布。这些点中的第 n 个被认为是函数值 $f(\lambda_n)$，高斯分布的良好的边缘化特征使本文能够以闭合的形式计算边界和条件。所得到的函数分布的性质完全由均值函数 $\mu:\Lambda\to\mathbb{R}$，$\mu(\lambda)=E[f(\lambda)]$ 和协方差函数（核函数）$k:\Lambda\times\Lambda\to\mathbb{R}$，$k(\lambda,\lambda')=E[(f(\lambda)-\mu(\lambda))(f(\lambda')-\mu(\lambda'))]$ 决定[7]。假设在输入集 $G=\{\lambda_1,\ldots,\lambda_t\}$ 和观测集的输出 $y=\{L(h_{\lambda_i},X_V)\}_{i=1}^t$ 上调节 $GP(\mu,k)$，其中 $y_n=f(\lambda_n)+\varepsilon$ 带有一个独立同分布的高斯噪声 $\varepsilon\sim\mathcal{N}(0,\sigma_n^2)$。测试点 λ_* 处的预测分布表示为

$$\hat{f}_*=k_*^T(K+\sigma_n^2 I)^{-1}y \tag{3}$$

$$V[f_*]=k(\lambda_*,\lambda_*)-k_*^T(K+\sigma_n^2 I)^{-1}k_* \tag{4}$$

其中：$k_*=[k(\lambda_1,\lambda_*),\ldots,k(\lambda_N,\lambda_*)]^T$，$K$ 是一个半正定矩阵 $[k(\lambda,\lambda')]_{\forall(\lambda,\lambda')\in(G\times G)}$。在每一次实验 t 中，GP 以全部的历史观测点值对集合 $\mathcal{H}=\{\lambda_i, L(h_{\lambda_i},X_V)\}_{i=1}^{t-1}$ 为条件评估 $f(\lambda_t)$。然后在利用平衡开发和勘探的采集函数的基础上，使用预测的后验均值和方差来选择下一组超参数。因为 GP 需要计算协方差矩阵的逆，因此它的计算复杂度为 $O(n^3)$。

2.2　对 GP 模型训练的优化

初始化 GP 时，采用一个零均值函数和一个 Matérn 5/2 协方差函数[7]。Matérn 5/2 kernel 表示为

$$K_{M_{52}}(\lambda,\lambda')=\theta_0\left(1+\sqrt{5r^2(\lambda,\lambda')}+\frac{5}{3}r^2(\lambda,\lambda')\right)exp\left\{-\sqrt{5r^2(\lambda,\lambda')}\right\} \tag{5}$$

其中 $r^2(\lambda,\lambda')=\sum_{d=1}^{D}\theta_d(\lambda_d-\lambda'_d)^2$，$\theta_d$ 为长度尺度，θ_0 为协方差振幅。与通常使用的平方指数内核相比，这个协方差函数产生二阶可微的样本函数，放宽了对后验概率密度函数平滑度的假设。本文介绍两种常用的训练 GP 超参数的方法，分别是最大化似然方法(maximum likelihood，ML)和 MCMC 方法，并对其中的 MCMC 方法进行优化。

2.2.1 Maximum Likelihood 方法

一旦获得一个新点，在这一点上评估确切的后验概率密度函数并更新 GP。在 GP 每次更新之后，通过最大化边际似然(marginal likelihood)来优化超参数(长度尺度 θ_d 和协方差振幅 θ_0)。log marginal likelihood 可以写成如下形式：

$$L = \log p\left(y|\lambda,\theta\right) = -\frac{1}{2}y^T\left(K+\sigma_n^2 I\right)^{-1}y - \frac{1}{2}\log\left|K+\sigma_n^2 I\right| - \frac{n}{2}\log 2\pi \quad (6)$$

训练目标为最大化针对训练样本的对数边际似然值，得到对应的超参数。并对超参数进行完全贝叶斯处理（仅由$\boldsymbol{\theta}$总结），并且求解边际似然函数的时候加入了 nugget term[8],协方差 K 应该替换为

$$K_v = \left(\left(1-v\right)r\left(\lambda_i-\lambda_j\right)+v\delta_{i,j}\right)_{i,j} \quad (7)$$

其中：nugget term 可以有效避免奇异解问题，使相关矩阵的条件数量适中，让最大似然估计更可靠。其计算复杂度为 O(n³)。

2.2.2 改进的 MCMC 方法

使用贝叶斯定理，可以得到模型参数$\boldsymbol{\theta}$的非标准化后验概率密度：

$$p\left(\theta|y\right) \propto p\left(y|\theta\right)p\left(\theta\right) \quad (8)$$

其中 $p(\theta)$ 表示 GP 模型参数的先验分布。

Metropolis Hasting（MH）通常用于从平稳分布生成样本的马尔可夫链[9]。假设马尔可夫链中的第 n 个样本为 θ_n，则其原生形式的 MH 首先从提议分布(proposal distribution) $q\left(\theta|\theta_n\right)$ 中提取一个随机候选并接受候选，其接受概率为：

$$\rho\left(\theta_n,\theta\right) = \min\left(1,\frac{q\left(\theta_n|\theta\right)p\left(\theta,y\right)}{q\left(\theta|\theta_n\right)p\left(\theta_n|y\right)}\right) \quad (9)$$

这意味着在每个建议的样本上都必须对代价高的后验概率密度函数式(8)上进行探测-即要运行的仿真模型。因为获得像后验概率密度函数这么精确的提议 $q\left(\theta|\theta_n\right)$ 是非常困难的，所以接受率通常很低，这导致了"拒绝不想要的提议点"花费了太多计算量。

为了提高 MH 的接受率,本文使用评估更快的 GP 代理来修改常用的提议分布，例如以 θ 为中心，协方差为 $\Sigma_p = \sigma_p^2 I$ 的高斯分布。候选样本 θ 是从建议分布中抽取的。不使用公式（9）直接测试这个候选点的接受程度，而是用概率 ρ_A 来测试这个候选点对 GP 代理 $p^*(\theta,y)$ 的接受程度：

$$\rho_A\left(\theta_n,\theta\right) = \min\left(1,\frac{q\left(\theta_n|\theta\right)p^*\left(\theta\,|\,y\right)}{q\left(\theta|\theta_n\right)p^*\left(\theta_n|y\right)}\right) \quad (10)$$

与式(9)相比，降低了很多计算成本。直观地说，通过过滤掉被精确概率密度函数拒绝的高概率的候选点来修改提议分布。只有被 GP surrogate 接受的候选点才会被接受评估,其概率为：

$$\rho_E\left(\theta_n,\theta\right) = \min\left(1,\frac{p\left(\theta|y\right)p^*\left(\theta_n\,|\,y\right)}{p\left(\theta_n\,|\,y\right)p^*\left(\theta|y\right)}\right) \quad (11)$$

通过这种方式提高了接受率，避免了以高概率被拒绝的提议，从而避免了不必要的模型拟合过程，其计算复杂度为 O(n²)+O(n)。这是在不牺牲采样精度的情况下实现的，因为最终的马尔可夫链是通过精确的后验概率密度函数生成的。由于篇幅有限，参考文献[14]用近似的方法详细讨论马尔可夫链的遍历性、收敛性和计算复杂度。

因为采用 Maximum likelihood 方法计算 GP 超参数的时候，需要进行求导等大量运算并且还可能在求解过程中陷入局部极值，因此本文采用改进的 MCMC 方法，该方法不仅使 GP 模型训练更快，而且在拟合模型的过程中不会出现局部最优解。两种方法的性能比较在下文实验部分给出。

2.3 采集函数

在基于模型的优化过程中，在每一次迭代 t 中，一个采样函数用于取样下一个点来评估。这个采集函数使用观察模型函数 $f(\lambda)$，并给每一组超参数一个量化值 $a\left(\lambda|f(\cdot)\right)$ 来平衡对新采样点的开发与勘探，以最大限度地找到全局最优解。而且本文的观测结果是 $\{\lambda_n,y_n\}_{n=1}^N$,其中 $y_n \sim N\left(f\left(\lambda_n\right),v\right)$ 和 v 是引入函数观测的噪声方差。用 $a:\Lambda\rightarrow R^+$ 表示的采集函数通过代理优化来确定下一步应评估$\boldsymbol{\Lambda}$中的哪个点 $\lambda_{nest} = \arg\max_\lambda a\left(\lambda|f(\cdot)\right)$，其中提出了几个不同的函数。一般来说，这些采集功能取决于先前的观察结果以及 GP 超参数;把这个依赖表示为 $a\left(\lambda;\{\lambda_n,y_n\},\theta\right)$。在高斯过程之前，这些函数完全依赖于模型的预测均值函数 $\mu\left(\lambda;\{\lambda_n,y_n\},\theta\right)$ 和预测方差函数 $\sigma\left(\lambda;\{\lambda_n,y_n\},\theta\right)$。在这个过程中，将 $\lambda_{best} = \arg\min_{\lambda_n} f\left(\lambda_n\right)$ 表示为最佳当前值。

2.3.1 常用的采集函数

a)GP upper confidence bound(GP-UCB)。通过寻找最大化 GP 的置信区间的点来完成的[7]:

$$\hat{\lambda} = arg\max_{\lambda}\left\{\mu\left(\lambda\right)+\beta^{1/2}\sigma\left(\lambda\right)\right\} \quad (12)$$

其中 $\mu(\lambda)$ 和 $\sigma(\lambda)$ 由 Sherman–Morrison–Woodbury 公式[7]评估。参数 β 使在样本空间的开发和探索保持平衡。式(12)使用 BOBYQA(bound optimization by quadratic approximation)进行优化[11]。

b)**Expected improvement**。可以选择在当前最好的情况下最大化期望增量（EI）。这在高斯过程下也有闭合形式：

$$a_{EI}\left(\lambda|\mu,\sigma\right)=E\left[\max\left(0,f\left(\lambda\right)-f\left(\lambda_{best}\right)\right)\right] \tag{13}$$

其中 λ_{best} 是目前为止基于观测集的最优解。由于其简单的形式，EI 使用标准的 black-box 优化算法进行优化[6]。

在本文中，本文重点讨论 EI 标准和本文提出的改进后的汤普森采集函数(Improved Thompson Sampling, ITS)，因为 GP-EI 已被证明比 GP-UCB 有更好的表现[6]，并且 GP-EI 不像 GP-UCB，它不需要自己调整参数。在实验部分，本文直接比较了 GP-EI、GP-ITS 以及 GP-UCB 的性能。

2.3.2 基于汤普森采样改进的采集函数

上面两种采集函数那样的探索性方法，在某些导致过度开发的情景下，这种对新观测点的采集往往是贪婪的[12]。虽然这种采集的特殊问题可以用类似 GLASSES 算法[15]的方式解决但这些都是难处理的。在本节中提出一种计算快速并且可以明确平衡勘探和开采的替代采集函数，是在汤普森抽样[13] (Thompson sampling)基础上进行改进的一种方法。对于任何新提出的 duel $[\lambda,\lambda']$，两个可能的结果 $\{0,1\}$ 对应于 λ 或 λ' 赢得了竞争(duel)[13]，两个结果的概率由 $\pi_{\tilde{f}}$ 给出。它遵循两步决策的竞争:

a)选择 λ： 首先，使用一个连续的汤普森采样[12]从模型生成一个样本 $\tilde{f}$，并通过对 Λ 进行积分来计算关联的 soft-Copland 分数。新 duel 的第一个元素 λ_{nest} 选择为

$$\lambda_{next}=\arg\max_{\lambda\in\Lambda}\int_{\Lambda}\pi_{\tilde{f}}\left(\left[\lambda,\lambda'\right];\mathcal{H}_j\right)d\lambda' \tag{14}$$

在 Copeland score 中的 $Vol(\Lambda)^{-1}$ 项在这里已经减小了，因为它不改变最优位置。这一步骤的目标是选择孔多塞胜者(Condorce Winner)时平衡勘探(exploration)和开发(exploitation)，这与汤普森抽样的做法是一样的：它可能选择一个接近当前 Condorcet 获胜者的点，但是该原则也允许探索其他地点，将决策建立在一个随机样本 $\tilde{f}$ 上。而且，收集的评估值越多，对 Condorce Winner 的选择就越贪婪。

b)选择 λ' :在给定 λ_{nest} 基础上，这个 duel 的第二个元素被选为在 λ_{nest} 的方向上使 $\sigma(f_*)$ 的方差最大化的位置。具体地说，λ_{nest} 被选择为

$$\lambda'_{nest}=\arg\max_{\lambda'_*\in\Lambda}V\left[\sigma\left(f_*\right)\middle|\left[\lambda_*,\lambda'_*\right],\mathcal{H}_j,\lambda_*=\lambda_{nest}\right] \tag{15}$$

这一步骤纯粹是在 λ_{new} 的方向上进行探索，其目标是找到信息丰富的比较，以便在前面步骤中确定的当前好位置上运行。

总之,改进后的汤普森取样法选择下一个 duel 为

$$\arg\max_{[\lambda,\lambda']}\alpha_{ITS}\left(\left[\lambda,\lambda'\right]\middle|\mathcal{H}_j\right)=\left[\lambda_{nest},\lambda'_{nest}\right] \tag{16}$$

其中: λ_{nest} 和 λ'_{nest} 是在上面定义的。该策略结合了一个点的选取和一个点的高概率，以及一个点的竞争结果相对于前一个点来说是不确定的。由于该采集函数可以平衡 exploration 和 exploitation，因此在超参数寻优过程中基本不会陷入局部极值。

2.4 改进的贝叶斯优化算法的实现

本文中使用拉丁超立方体抽样(Latin hypercube sampling, LHS）方法初始化观测集。经过改进的算法实现如下：

算法 1：基于改进的贝叶斯算法的 CNN 超参数优化算法(GP ITS IMCMC）

Input: X_T and X_V , some training and validation datasets, N the number of initial points , A the CNN algorithm, and L the loss function

Output: Return the best model of CNN

$\mathcal{H}_0\leftarrow\left\{\left[\lambda_j,\lambda'_j\right],L\left(h_{\lambda_j},X_V\right)\right\}_{j=1}^{N}$ by LHS

for $t\in 1,\ldots,T$ **do**

　$f(\lambda)\leftarrow GP(\mathcal{H}_{t-1})$ and learn $\pi_{f,t-1}(\lambda)$ //Fit GP

　Compute the acquisition for duels α .

　Next duel: $\left[\lambda_t,\lambda'_t\right]=\arg\max\alpha\left(\left[\lambda,\lambda'\right]\right)$

　$m_{\lambda_t}\leftarrow A\left(\left[\lambda_t,\lambda'_t\right],X_T\right)$ //Train model

　$l_{\lambda_t}\leftarrow L\left(m_{\lambda_t},X_V\right)$ //Compute validation loss

　$\mathcal{H}_t\leftarrow\mathcal{H}_{t-1}\cup\left\{\left(\left[\lambda_t,\lambda'_t\right],l_{\lambda_t}\right)\right\}$ //Update observations

end for

return $\arg\min_{m\in\mathcal{H}_T}L\left(m,X_V\right)$

3 实证分析

在本节中，通过实证分析了本文提出的算法，并对现在常用的超参数优化进行比较。本文进行比较的算法主要有 Bergstra 等人提出的 TPE 算法和 Jasper Snoek 等人提出的 GP EI MCMC 算法。首先使用 Branin-Hoo 函数简单测试本文算法的性能;然后使用 CIFAR-10、MRBI(rotated MNIST with background images) 和 SVHN（Street View House Numbers Dataset）数据集测试本文算法对 CNN 超参数的优化性能，并与同类优化算法比较得出结论。

3.1 实验环境

众所周知，神经网络和深度学习方法需要仔细调整大量超参数。多层卷积神经网络就是这样一个模型的一个例子，如 Saxe 等人所证明的那样，对体系结构和超参数进行彻底的探索是非常有益、有必要的。本文研究了一个与 Snoek 和 Domhan 使用的 cuda-convnet 相同架构的卷积神经网络[16]。Snoek 和 Domhan 使用的超参数的搜索空间不同，本文使用类似于 Snoek 等人的搜索空间。其中 6 个超参数用于随机梯度下降，两个超参数用于响应归一化层(response normalization layers)，搜索空间如表 1 所示。Snoek 等人的两个超参数被排除在本文的实验之外：由于 Caffe 框架的限制，响应归一化层的宽度被排除；由于与动态资源分配不相容，所以 epochs 数目被排除。

表 1 三层卷积神经网络的超参数及其范围

Hyper-parameter	Scale	Min	Max
Learning Parameters			
Initial Learning Rate	log	$5*10^{-5}$	5
Conv1 l_2 Penalty	log	$5*10^{-5}$	5
Conv2 l_2 Penalty	log	$5*10^{-5}$	5
Conv3 l_2 Penalty	log	$5*10^{-5}$	5
FC4 l_2 Penalty	log	$5*10^{-5}$	5
Learning Rate Reductions	integer	0	3
Local Response Normalization			
Scale	log	$5*10^{-6}$	5
Power	linear	$1*10^{-2}$	3

3.2 实验结果

3.2.1 Branin-Hoo 函数

利用 Branin-Hoo 函数测试比较 MML(maximum marginal likelihood，MML)方法和改进后的 IMCMC(improved MCMC)方法训练 GP 的性能，同时比较标准的方法和 TPE[5]方法的性能（按函数计算的次数表述），如图 1 所示。Branin-Hoo 函数定义在 $0 \le x_1 \le 15, 0 \le x_2 \le 15, x \in \mathbb{R}$ 上，是贝叶斯优化技术的共同标准[6]。在 Branin-Hoo 中，GP-EI 和 GP-ITS 明显优于 TPE 和 GP-UCB，IMCMC 方法优于 MML。

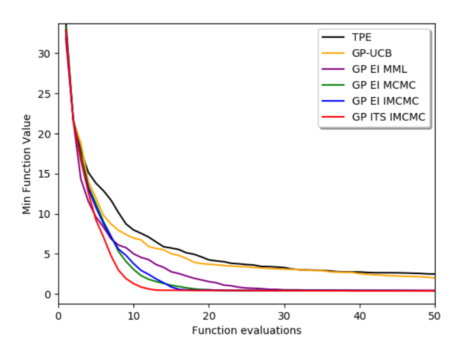
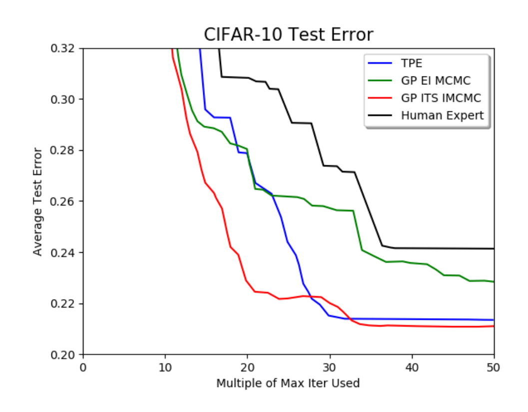


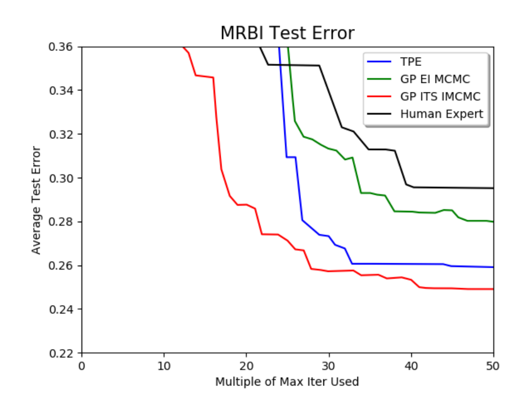
图 1 在 Branin-Hoo 函数上比较 GP ITS IMCMC 和一些标准方法

3.2.2 Alex Krizhevsky's CNN 框架

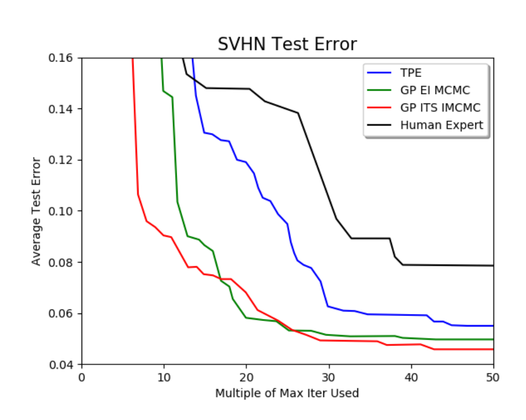
本文使用三个图像分类数据集：CIFAR-10、MRBI（The MNIST Rotated Background Images Dataset）和 SVHN（The Street View House Numbers Dataset）。CIFAR-10 和 SVHN 包含 32x32 的 RGB 图像，而 MRBI 包含 28x28 的灰度图像。每个数据集分为训练、验证和测试集：CIFAR-10 有 40000，10000 和 10000 个实例； SVHN 分别有 600000，6000，26000； MRBI 分别有 10000，2000，50000 个样本用于训练、验证和测试。对所有数据集进行图像归一化和均值削减预处理。本文为 CIFAR-10 和 MRBI 设置 max_iter 为 300（对于 CIFAR，这对应于训练集上的 75 个 epochs），而对于 SVHN，由于其较大的训练集，最大迭代 max_iter 被设置为 600。在以下的所有图表中，该算法使用 Validation Error 进行最优超参数的选取，使用 Test Error 评估最优超参数的性能。



(a)CIFAR-10 测试误差结果图



(b)MRBI 测试误差结果图



(c)SVHN 测试误差结果图

图 2 比较不同超参数优化算法在 CIFAR-10、MRBI 和 SVHN 数据集上的平均测试误差

本文使用 GP ITS IMCMC 对 CIFAR-10、MRBI 和 SVHN 测试集上的 CNN 的 8 个超参数进行了优化，并在随机初始化运行中报告平均测试误差(Average Test error). 并与使用 TPE、GP EI MCMC 方法以及人类专家[10] (Human Expert)凭经验获得的最佳参数获得的平均结果进行对比。结果显示在图 2 中，从图（a）（b）中可以看出：在 CIFAR-10 测试集和 MRBI 测试集中，TPE 性能优于 GP EI MCMC 和 Human Expert，GP ITS IMCMC 性能优于 TPE；从图（c）中可以看出：在 SVHN 测试集中，GP EI MCMC 性能优于 TPE 和 Human Expert，GP ITS IMCMC 性能优于 GP EI MCMC。从总体上看，GP ITS IMCMC

找到的最佳参数比 TPE 方法和 GP EI MCMC 方法找到的超参数平均性能高出了 1%以上，比 Human Expert 找到的超参数平均性能高出了 3%以上。

4　结束语

本文提出了一种基于改进的贝叶斯优化算法的卷积神经网络超参数优化算法。该方法利用高斯过程回归对超参数进行建模，把一种以改进的汤普森采样方法作为采集函数，并且使用改进的马尔可夫链蒙特卡洛算法(IMCMC)对 GP 超参数求解过程进行加速。该方法适用于各种不同卷积神经网络的超参数优化。实验结果表明该方法可以比目前其他常见的优化算法表现出更好的性能。下一步工作的重点在于将 GP ITS IMCMC 方法进行并行化来提高卷积神经网络超参数优化算法的收敛速度。

参考文献：

[1] Breuel T M. The effects of hyperparameters on SGD training of neural networks [J]. arXiv preprint arXiv: 1508. 02788, 2015.

[2] Mockus J. Bayesian approach to global optimization: theory and applications [M]. Springer Science & Business Media, 2012.

[3] Jones D R. A taxonomy of global optimization methods based on response surfaces [J]. Journal of global optimization, 2001, 21 (4): 345-383.

[4] Klein A, Falkner S, Bartels S, *et al.* Fast bayesian optimization of machine learning hyperparameters on large datasets [J]. arXiv preprint arXiv: 1605. 07079, 2016.

[5] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for Hyper-Parameter Optimization [C]// Proc. of NIPS. 2011.

[6] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms [C]// Proc of NIPS. 2012.

[7] Rasmussen C E, Williams C K I. Gaussian processes for machine learning [M]. Cambridge: MIT press, 2006.

[8] Pepelyshev A. The role of the nugget term in the Gaussian process method [M]// Advances in Model-Oriented Design and Analysis. Physica-Verlag HD, 2010: 149-156.

[9] Kapli P, Lutteropp S, Zhang J, *et al.* Multi-rate Poisson tree processes for single-locus species delimitation under maximum likelihood and Markov chain Monte Carlo [J]. Bioinformatics, 2017, 33 (11): 1630-1638.

[10] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images [EB/OL]. (2009-04-08) . http://www. cs. toronto. edu/~kriz/learning-features-2009-TR. pdf.

[11] Powell M J D. Developments of NEWUOA for minimization without derivatives [J]. IMA Journal of Numerical Analysis, 2008, 28 (4): 649-664.

[12] Hernández-Lobato J M, Hoffman M W, Ghahramani Z. Predictive entropy search for efficient global optimization of black-box functions [C]// Advances in Neural Information Processing Systems. 2014: 918-926.

[13] Wu H, Liu X. Double thompson sampling for dueling bandits [C]// Advances in Neural Information Processing Systems. 2016: 649-657.

[14] Christen J A, Fox C. Markov chain Monte Carlo using an approximation [J]. Journal of Computational and Graphical statistics, 2005, 14 (4): 795-810.

[15] González J, Osborne M, Lawrence N. GLASSES: Relieving the myopia of Bayesian optimisation [C]// Artificial Intelligence and Statistics. 2016: 790-799.

[16] Domhan T, Springenberg J T, Hutter F. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves [C]// Prod of IJCAI. 2015: 3460-3468.